*Proc. of the 8th International Symposium on Geotechnical Safety and Risk (ISGSR)*
*Edited by* Jinsong Huang, D.V. Griffiths, Shui-Hua Jiang, Anna Giacomini and Richard Kelly
 *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-5182-7_08-011-cd

ISGSR 2022

# An Unsupervised Framework for Mud Pumping Detection and Severity Analysis Using In-Service Train Data in Railway Track

Cheng Zeng[1], Jinsong Huang[2], Jiawei Xie[3]

[1]Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia.
E-mail: cheng.zeng@uon.edu.au
[2]Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia.
E-mail: jinsong.huang@newcastle.edu.au
[3]Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia.
E-mail: jiawei.xie@uon.edu.au

**Abstract:** Using machine learning techniques to analyze the monitoring data collected from in-service trains can help the infrastructure manager to detect and localize mud pumping defects automatically. However, most studies treat defect detection tasks in a supervised manner (e.g., classification), which rely heavily on manual data processing to label unhealthy conditions (such as mud pumping) for training. But a majority of measurement data actually represents a healthy condition. Supervised classification models require considerable efforts to balance the training dataset; otherwise, the models tend to perform poorly for the minority class. Unsupervised anomaly detection can handle extremely imbalanced dataset because no label information is needed in the unsupervised mode. This study proposes an unsupervised framework for mud pumping detection and severity analysis using in-service train data. The framework is based on a long short-term memory (LSTM) autoencoder and deep embedding clustering (DEC). The proposed framework is implemented on a three-year dataset collected from a section of railroads in Australia. The results show that the model can detect 5 out of 6 mud pumping events and identify their severity.

Keywords: Unsupervised learning; imbalanced dataset; autoencoder; deep embedding clustering; mud pumping.

## 1 Introduction

Mud pumping is one of the severe substructure problems faced by the rail industry. The increasing development of mud pumping will deteriorate the track condition and greatly impact the safe operation of trains. Therefore, it is important to detect mud pumping at early stage to avoid excessive deformation, track misalignment, and potential derailment. Recently, on-board sensor-based track condition monitoring has been widely applied. By mounting robust sensors on in-service trains, an almost continuous and real-time track monitoring is possible for entire rail networks in a timely and cost-efficient manner. It has been proved that dynamic responses collected by axle-box accelerometers provided valuable insights into the vehicle-track interaction and could be linked to track degradation such as corrugation and squats (Westeon et al. 2007, Molodova et al. 2014, Rapp et al. 2019). Researchers have developed many on-board sensor data-based methods for track defects detection. Jamshidi et al. (2018) proposed a deep convolutional neural network to detect squats on rail surface using axle-box accelerations. Zeng et al. (2021) proposed an LSTM-based network using in-service train data to predict the location where mud pumping is most likely to occur. However, most researchers treat defect detection tasks in a supervised manner (e.g., classification), which relies heavily on manual data processing to label unhealthy conditions (such as mud pumping) for training. But a majority of measurement data actually represents a healthy condition. Supervised classification models require considerable efforts to balance the training dataset; otherwise, the models tend to perform poorly as imbalanced datasets induce a bias in favor of the majority class.

Unsupervised learning algorithms that do not require label information seem to be a promising solution to this issue. A few studies have developed unsupervised deep learning algorithms to identify track defects using on-board sensor data. Yuan et al. (2021) proposed an unsupervised detection algorithm for light rail squat localization. But, their study was only based on simulated dynamic responses from vehicle-track coupled model and in-lab scale model. Niebling et al. (2020) conducted initial research of using gaussian mixture model to detect track irregularities from axle-box acceleration. Although their data was collected from in-service train in the field, their study pointed out that their method cannot lead to reliable results because no validation with the real occurred defects was considered. Thus, unsupervised learning algorithms based on real in-service train data have not been fully exploited to detect defects on tracks. There are several reasons for this lack of exploration, and they all present major challenges. First, one widely used unsupervised algorithm is distance-based clustering,

such as K-mean clustering. However, the traditional distance-based clustering methods fail to work effectively when dealing with in-service train data, which is high dimensional and in the format of time sequences. Second, in real-world applications, it would be ideal if a defect detection algorithm could provide operators with knowledge of defect severity. But the probability of clustering may not be relevant to the severity because it is only the measure of the distance between the sample and the cluster centroid.

This paper proposes an unsupervised framework to jointly consider the aforementioned issues. The framework is based on a long short-term memory (LSTM) autoencoder and deep embedding clustering (DEC). Specifically, an LSTM autoencoder is introduced to learn a mapping between input sequences and their latent representations, thereby reducing the high-dimensionality and learning temporal dependences of the input sequences. Subsequently, DEC, a method to integrate the learned mapping and K-mean clustering, is applied for mud pumping detection. Finally, the LSTM autoencoder is reused to reconstruct the input sequences and the reconstruction errors are further utilized for severity analysis. The assumption is that the mud pumping instances are difficult to be reconstructed from the latent representations and thus have large reconstruction errors. Mud pumping instance with larger reconstruction error is assumed to be more severe. The proposed framework is implemented on a three-year dataset collected from a section of railroads in Australia. The results show that the model can detect 5 out of 6 mud pumping events and identify their severity.

## 2 Methodology

The proposed LSTM-DEC framework for mud pumping detection and severity analysis mainly includes LSTM autoencoder to reduce the high-dimensionality of input, deep embedding clustering to detect mud pumping, and reconstruction errors for severity analysis.

### 2.1 LSTM autoencoder

In-service train data is high-dimensional time sequences with temporal dependences. Directly applying traditional clustering methods, such as K-means, to in-service train data may have low performance because traditional methods struggle to deal with non-linear relations among those time sequences. An autoencoder is a neural network that can be used to compress high-dimensional input to lower-dimensional representations and still keep the main information in data. By introducing recurrent neural networks into the autoencoder, the temporal information in the sequential data can be well captured. Thus, an LSTM autoencoder is proposed to learn the mapping between input and latent representation.

LSTM autoencoder consists of encoder $E$ and decoder $D$. In particular, given the input $x$, the encoder compresses $x$ to obtain a low dimensional representation $z = E(x)$. The decoder reconstructs this representation to give the output $\hat{x} = D(z)$. The autoencoder is trained by minimizing the reconstruction error:

$$\mathrm{L}_R = \mathrm{E}_{x_{1:T}\sim p}\left[\sum_{t=1}^{T}\left\|x_t - D\left(E\left(x_t\right)\right)\right\|_2\right] \tag{1}$$

where $T$ is the length of time steps.

### 2.2 Deep embedded clustering (DEC)

After having learned mapping, there are two ways to apply cluster methods for mud pumping detection. One is applying clustering to the latent representation directly. Another is treating the learned mapping as an initial estimation and then jointly optimizing the mapping and clustering simultaneously, called deep embedded clustering (DEC) (Xie et al. 2016). During the training process of DEC, the cluster assignments are optimized in a self-supervised manner. It has been shown that DEC can obtain state-of-the-art clustering results (Guo et al. 2017, Asadi and Regan 2019). Thus, DEC is applied for the clustering task for mud pumping detection in this study. The clustering objective and optimization algorithm proposed by (Xie et al. 2016) is used. The detailed mathematical formulation of DEC can be retrieved by (Xie et al. 2016).

### 2.3 Reconstruction error

Given the clustering results, the next task is to analyze the severity of detected mud pumping events. A common practice is to take the probability of clustering a sample to mud pumping cluster as a measure score, that is, the higher the probability, the more likely the sample is to be mud pumping and thus the more severe. But the probability of clustering may not be relevant to the severity because it is only the measure of the distance between the sample and the cluster centroid. One of the applications of autoencoder is identifying anomaly samples through reconstruction errors. The intuition is that the learned latent representations are enforced to learn important regularities of the data to minimize reconstruction errors; defect instances are difficult to be reconstructed from the resulting representations and thus have large reconstruction errors. Inspired by this, mud

pumping event with larger reconstruction error is assumed to be more severe. The LSTM autoencoder is then reused to reconstruct the input and calculate reconstruction errors for severity analysis.

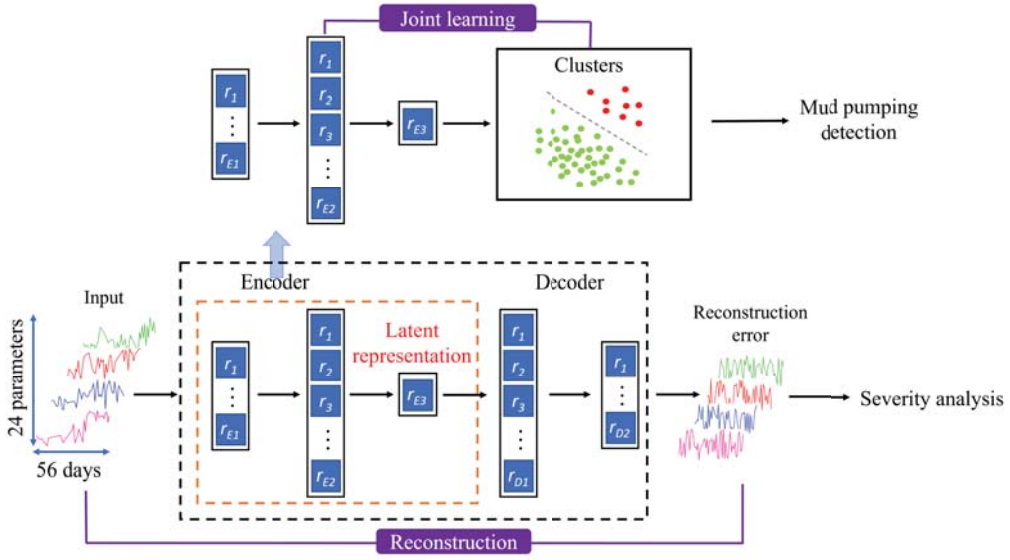


**Figure 1.** Framework for mud pumping detection and severity analysis.

The whole framework is shown in Figure 1. The framework starts with pre-training an LSTM autoencoder to provide a reversible mapping between input and latent representations, thereby reducing the high dimensionality of the clustering space. Then, given a pre-trained encoder and initial cluster, a joint learning is performed to find a deep embedding clustering. This learned deep embedding clustering can separate the dataset into two clusters, where the minority clusters are considered as mud pumping events. Finally, given the clustering results, the mud pumping events are reconstructed based upon the pre-trained LSTM autoencoder. The outputs of autoencoder are the lossy reconstruction of the input. The reconstruction error is then used to analyze the severity of mud pumping events.

## 3 Results and Discussion

### 3.1 Dataset

Dataset available for the current study consists of in-service train data and track characteristic data including train speed and annual tonnage. The dataset is collected from a section of railroads (up to 120 km) in Australia for three years from 2018 to 2021. The in-service trains collect sampling data along the track with daily service. The collected sampling data is pre-processed into 22 parameters at 1-meter interval with corresponding locations. The details of the 22 parameters (corresponding to parameter No. 1-22) are shown in Table 1. A total of 24 parameters will be used for mud pumping detection as can be seen from Table 1.

As suggested by Mohammadi et al. (2019) and Li et al. (2014), the changes of parameters over time should be considered when predicting rail conditions. Data used in this study can be processed as time sequences to preserve the changes over time. As in-service trains collect data almost every day, the time interval is set to be one day. Based on previous data processing methods by Zeng et al. (2021), all in-service train parameters and train speed are processed as time sequences. As the annual tonnage of a segment does not vary with time, there is no need for processing.

The data, which is within 10 m before and after the mud pumping location and recorded before the occurred date, is labeled as a ‘mud pumping’ sample. A time window is introduced to determine how many days of data should be used for model development. According to a previous study (Zeng et al. 2021), the time window is set to be 56 days in this study. The data outside 10 m of the mud pumping location is labeled as ‘non-mud pumping’ sample. It is worth mentioning that the labels are used only to evaluate the efficiency of the framework and they are not used by the unsupervised model. The time window of non-break is also 56 days that are randomly selected and consecutive. Each sample is the time sequences including the changes of 24 parameters over 56 days, with the shape of [56×24].

### 3.2 Mud pumping detection

To evaluate the performance of the proposed models, a real-life validation is carried out. One of the most common approaches is to randomly select 70% of samples for training and the rest 30% for testing. In this study, the train and test split is defined based on time. All the samples collected before 2020 are used for training, while those collected after 2020 are used for testing purposes. The split based on time will evaluate the model

performance more robustly, where the model is trained on only past data and tested out on future data. In the training data, the number of samples in the mud pumping class and non-mud pumping class are 10 and 433, respectively. In the testing data, the number of samples in the mud pumping class and non-mud pumping class are 6 and 250, respectively.

**Table 1.** The details of parameters.

| No. | Parameters | Details |
|---|---|---|
| 1 | Va | Maximum value of vertical acceleration (Va) |
| 2 | Va_L | Maximum value of Va (front bogie) |
| 3 | Va_R | Maximum value of Va (rear bogie) |
| 4 | Vastd_F(L) | The standard deviation of left-side Va (front bogie) |
| 5 | Vastd_F(R) | The standard deviation of right-side Va (front bogie) |
| 6 | Vastd_R(L) | The standard deviation of left-side Va (rear bogie) |
| 7 | Vastd_R(R) | The standard deviation of right-side Va (rear bogie) |
| 8 | SD | Maximum value of suspension displacement (SD) |
| 9 | SD_L | Maximum value of SD (front bogie) |
| 10 | SD_R | Maximum value of SD (rear bogie) |
| 11 | SDdif_F | The deviation of SD between left-side and right-side frame (front bogie) |
| 12 | SDdif_R | The deviation of SD between left-side and right-side frame (rear bogie) |
| 13 | SDavg_F | The mean value of SD between left-side and right-side frame (front bogie) |
| 14 | SDavg_R | The mean value of SD between left-side and right-side frame (rear bogie) |
| 15 | Twist1 | Track twist (2 m chords) |
| 16 | Twist2 | Track twist (14 m chords) |
| 17 | CRV | Track curvature |
| 18 | Pro(L) | Vertical profile of left rail |
| 19 | Pro(R) | Vertical profile of right rail |
| 20 | Align | GPS alignment |
| 21 | BCR | Brake cylinder pressure |
| 22 | Force | In-train forces |
| 23 | Tonnage | Annual tonnage |
| 24 | Speed | Train speed |

Since the dataset is imbalanced, using the standard accuracy as an evaluation metric may lead to a prediction model looking promising with high accuracy but failing to be valid in detecting mud pumping. Hence, false prediction of mud pumping and false prediction of non-mud pumping are introduced for measuring the performance of the model. In addition, balanced accuracy is introduced as a compact evaluation metric to reflect the general performance of prediction models. Balanced accuracy is especially useful when the test dataset is imbalanced.

$$\text{False prediction of mud pumping} = \frac{FN}{P} \tag{5}$$

$$\text{False prediction of non-mud pumping} = \frac{FP}{N} \tag{6}$$

$$\text{Balanced accuracy} = 1 - \frac{1}{2} \times \left( \frac{FN}{P} + \frac{FP}{N} \right) \tag{7}$$

where $P$ indicates the total number of real mud pumping samples and $FN$ denotes the number of mud pumping falsely predicted as non-mud pumping. $N$ denotes the total number of real non-mud pumping samples and $FN$ denotes the number of non-mud pumping samples falsely predicted as mud pumping.

The proposed models are trained via mini-batch stochastic gradient descent with the Adam optimizer. The size of the minibatch is 64. The learning rate is set to be 0.001. The number of epochs is set as 100 because the loss is stable according to multiple experiments. After training, the proposed model achieves the false prediction of mud pumping of 16.7%, the false prediction of non-mud pumping of 17%, and the balanced accuracy of 83.2% on testing data.

To gain insight into what the LSTM autoencoder has learned, the latent representations are investigated. Figure 2 shows the 3D latent representations learned by the autoencoder for the testing data. The points plotted in Figure 2 are latent representations generated by the testing data containing 6 mud pumping and 250 non-mud pumping samples, where the star denotes mud pumping, and the circle denotes non-mud pumping. In latent representations, two clusters can be observed, with one gathering a majority of samples and the other with much fewer samples. In Figure 2, there is a small gap between the oval-shaped big cluster and the small one locates

closer to the vertical axis. The representations demonstrate that a certain data structure has been learned by the autoencoder, the learned features are quite discriminative.

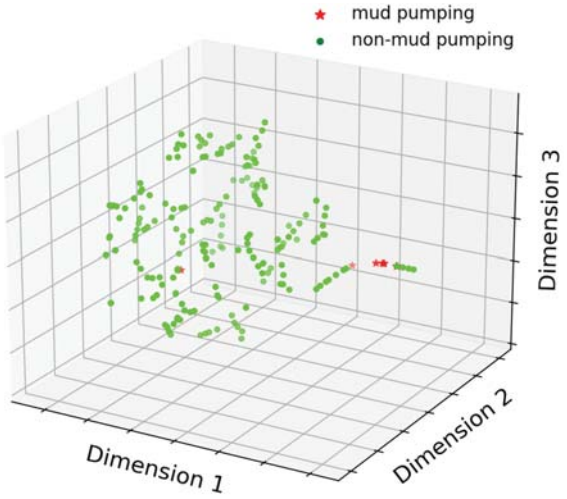


**Figure 2.** The learned 3D latent representations of testing data.

To demonstrate the superiority of the proposed model, the previous LSTM-based supervised classification model proposed by Zeng et al. (2021) is applied to the same datasets for comparison. Since the dataset is imbalanced, it is necessary to make efforts to balance the data before training. Synthetic minority-oversampling technique (SMOTE) is one of the most widely used algorithms to generate synthetic samples to overcome the data imbalance issue (Chawla et al. 2002, Suh et al. 2021). It generates synthetic samples between each real sample and its k-nearest neighbors. In this study, to handle the data imbalance in supervised training, SMOTE is used to generate samples for mud pumping events. Besides, the commonly used anomaly detection algorithm, LSTM based-autoencoder (LSTM-AE), is selected as an unsupervised learning algorithm for comparison. In addition, the LSTM-based supervised classification model with no oversampling method applied is also included. Overall, 4 prediction models are constructed.

The results are compared in Table 2, where the best performances for each metric are highlighted in bold. It is noted that LSTM-based supervised model with no oversampling achieves the lowest false prediction of non-mud pumping of 0.0% but with a high false prediction of mud pumping of 83.3% and low balanced accuracy of 58.4%. Only 1 of 6 mud pumping events can be detected correctly. While with the help of SMOTE, LSTM based supervised model performs better than that without. But more than half of the mud pumping samples are still undetected. This demonstrates that supervised models are struggled to deal with extremely class imbalance. Unsupervised anomaly detection can handle extremely imbalanced dataset because no label information is needed in the unsupervised mode. This also can be confirmed from Table 2 that both two unsupervised models outperform supervised models with a higher balanced accuracy of 79.4% and 83.2%, respectively. In general, the proposed model achieves the lowest false prediction of mud pumping of 16.7% and the highest balanced accuracy of 83.2%.

**Table 2.** Comparative results. Best results for corresponding metric are bold.

| Training mode | Detection model | False prediction of mud pumping (The lower the better) | False prediction of non-mud pumping | Balanced accuracy (The higher the better) |
|---|---|---|---|---|
| supervised | LSTM-None | 83.3% | 0.0% | 58.4% |
| | LSTM-SMOTE | 66.7% | 0.0% | 66.7% |
| unsupervised | LSTM-AE | 33.3% | 8.0% | 79.4% |
| | LSTM-DEC | 16.7% | 17.0% | 83.2% |

### 3.3 Severity analysis

To analyze mud pumping severity, the testing data is fed into the pre-trained autoencoder. The outputs of autoencoder are the lossy reconstruction of the input. The reconstruction error then is used to measure the severity of mud pumping, that is, the larger the reconstruction error the more severe the mud pumping. Figure 3 shows the reconstruction errors of testing data, where the star represents mud pumping sample, and the circle represents the non-mud pumping sample. Each mud pumping sample is marked with its affected length. It can be seen from Figure 3 that the affected length of mud pumping events is indeed highly correlated with the reconstruction errors. The longest affected length is associated with the biggest reconstruction error, while the shortest affected length is associated with the smallest reconstruction error. It can be concluded that

reconstruction errors can provide insight into the severity of mud pumping events. This helps infrastructure managers perform maintenance priorities according to the severity of mud pumping events.

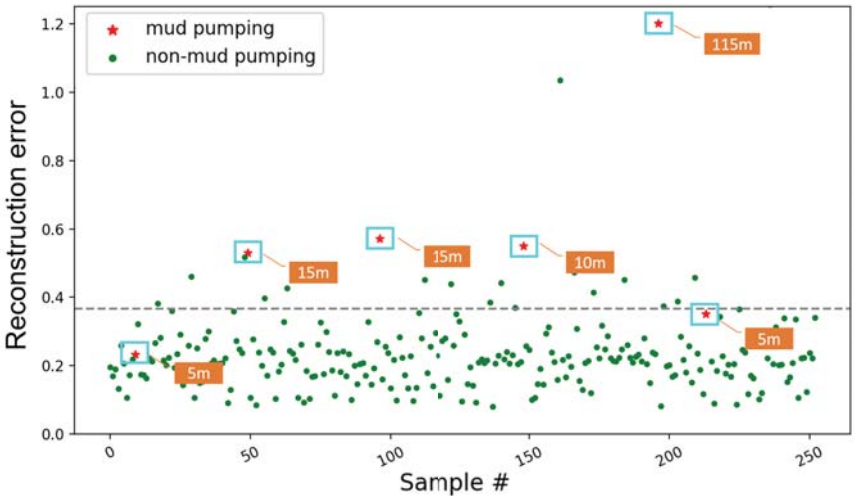


**Figure 3.** Severity analysis of testing data.

## 4 Conclusion

This paper proposes an unsupervised framework for mud pumping detection and severity analysis. The framework is based on a long short-term memory (LSTM) autoencoder and deep embedding clustering (DEC). An LSTM autoencoder is introduced to learn a mapping between input sequences and their latent representations. DEC is applied for mud pumping detection. The LSTM autoencoder is then reused to reconstruct the input sequences and the reconstruction error is further utilized for severity analysis. The proposed framework is implemented on a three-year dataset collected from a section of railroads in Australia. The results show that the model can detect 5 out of 6 mud pumping events and identify their severity.

## References

Asadi, R. and A. Regan, 2019. Spatio-temporal clustering of traffic data with deep embedded clustering. Proceedings of the 3rd ACM SIGSPATIAL *International Workshop on Prediction of Human Mobility*.

Chawla, N. V., et al., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16: 321-357.

Guo, X., et al., 2017. Improved deep embedded clustering with local structure preservation. *Ijcai*.

Jamshidi, A., et al., 2018. A decision support approach for condition-based maintenance of rails based on big data analysis. *Transportation Research Part C Emerging Technologies* 95: 185-206.

Li, H., et al., 2014. Improving rail network velocity: a machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies* 45: 17-26.

Mohammadi, R., et al., 2019. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. *Transportation Research Part C: Emerging Technologies* 102: 153-172.

Molodova, M., et al., 2014. Automatic detection of squats in railway infrastructure. *IEEE transactions on intelligent transportation systems* 15(5): 1980-1990.

Niebling, J., et al., 2020. Analysis of Railway Track Irregularities with Convolutional Autoencoders and Clustering Algorithms. *European Dependable Computing Conference, Springer*.

Rapp, S., et al., 2019. Track-vehicle scale model for evaluating local track defects detection methods. *Transportation Geotechnics* 19: 9-18.

Suh, S., et al., 2021. CEGAN: Classification Enhancement Generative Adversarial Networks for unraveling data imbalance problems. *Neural Networks* 133: 69-86.

Westeon, P., et al., 2007. Monitoring vertical track irregularity from in-service railway vehicles. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 221(1): 75-88.

Xie, J., et al., 2016. Unsupervised deep embedding for clustering analysis. *International conference on machine learning, PMLR.*

Yuan, Z., et al., 2021. An unsupervised method based on convolutional variational auto-encoder and anomaly detection algorithms for light rail squat localization. *Construction and Building Materials* 313: 125563.

Zeng, C., et al., 2021. Prediction of mud pumping in railway track using in-service train data. *Transportation Geotechnics* 31: 100651.